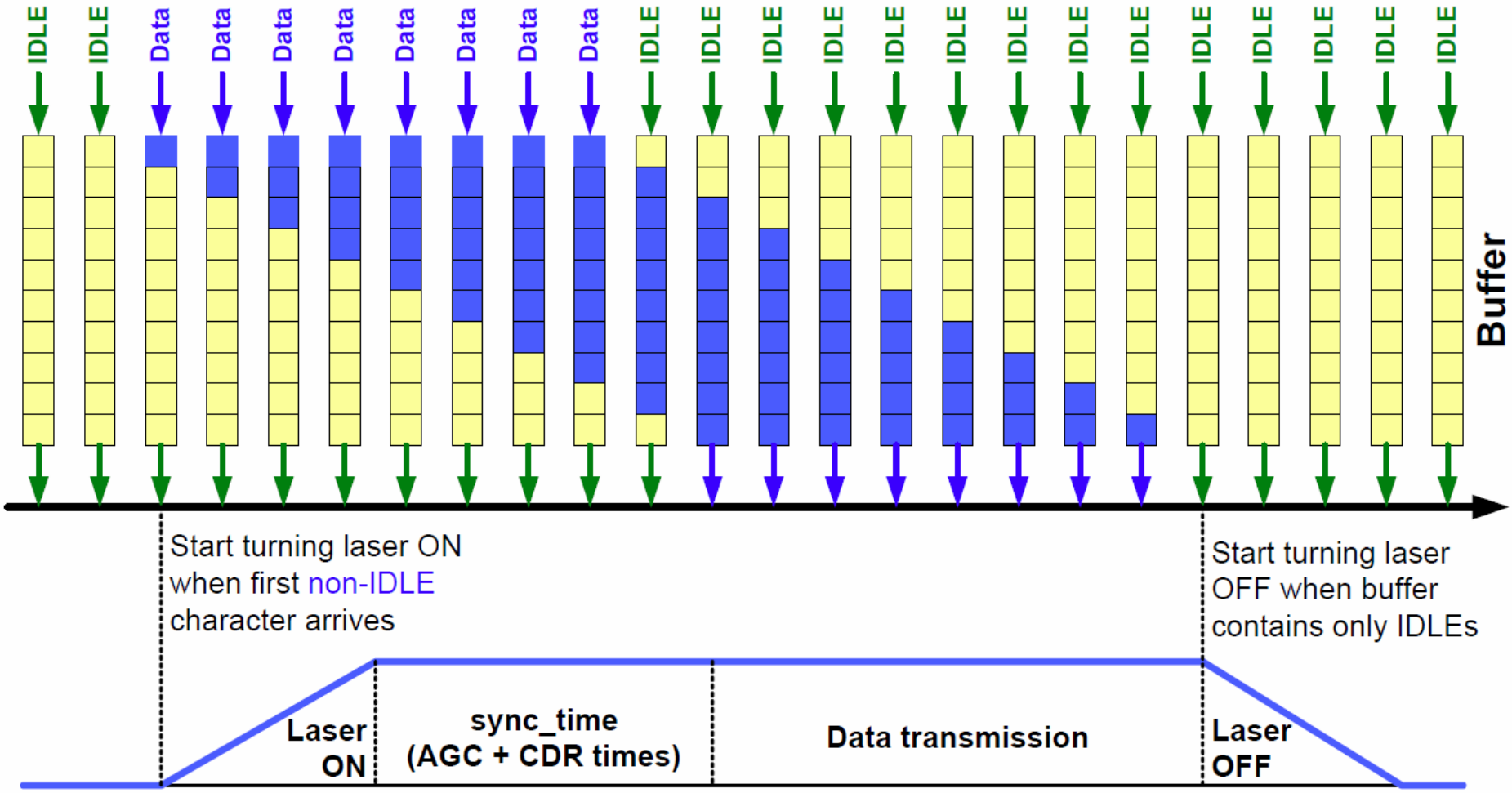
Data Detector:
The history, the concept, and the needed adaptation for 802.3ca

Glen Kramer (Broadcom Ltd.)

History

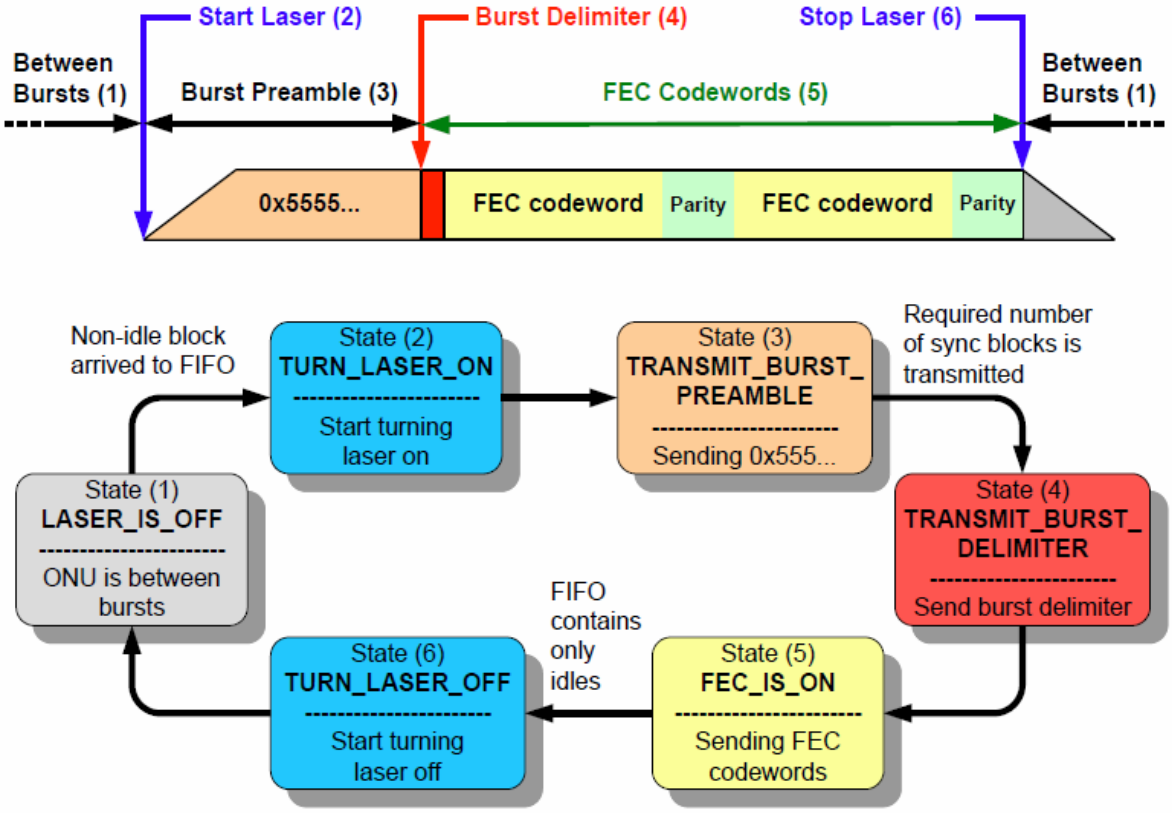
- Data Detector was suggested in September 2003 (802.3ah) to eliminate the LaserControl signal from MAC Control to PMD, which 802.3 considered a layering violation.
- It underwent minor modifications in March 2007 (802.3av) to adapt it to 64b/66b line coding and stream-based FEC.
- It has a simple concept and provides an easy-to-understand behavioral model to the implementers.

Illustration of the approach



Conceptual Diagram

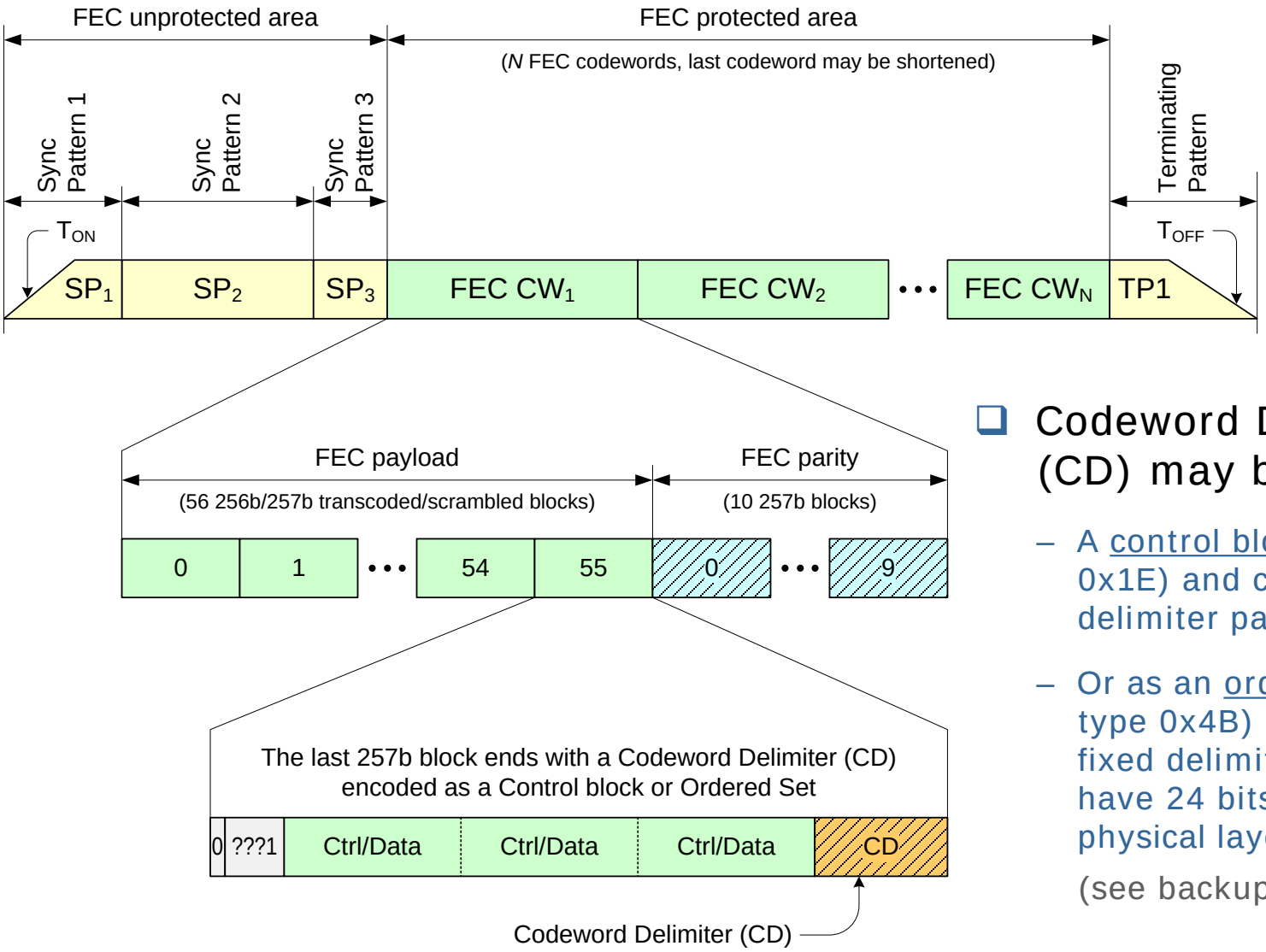
- There are 6 stages in burst's "lifetime"
  1. Between bursts
  2. Turning laser on
  3. Sending preamble
  4. Sending delimiter
  5. Sending FEC-protected data
  6. Turning laser off
- The stages always follow in the same order



Data Detector for 802.3ca

- There are no fundamental changes in burst composition in 802.3ca vs 802.3av. The changes are relatively minor and the existing Data Detector state diagram can be easily adapted to 802.3ca.
- What has changed:
  1. Burst preamble consists of three programmable patterns of programmable lengths.
  2. No need to resynchronize the scrambler (two sacrificial idle blocks at the front of every burst are gone)
  3. FEC Codeword size is different. Last codeword in a burst may be shortened.
  4. Line-coding block size is different (257b vs. 66b)
  5. Last 257b block in FEC payload includes the Codeword Delineation Marker (CDM)

Burst Composition

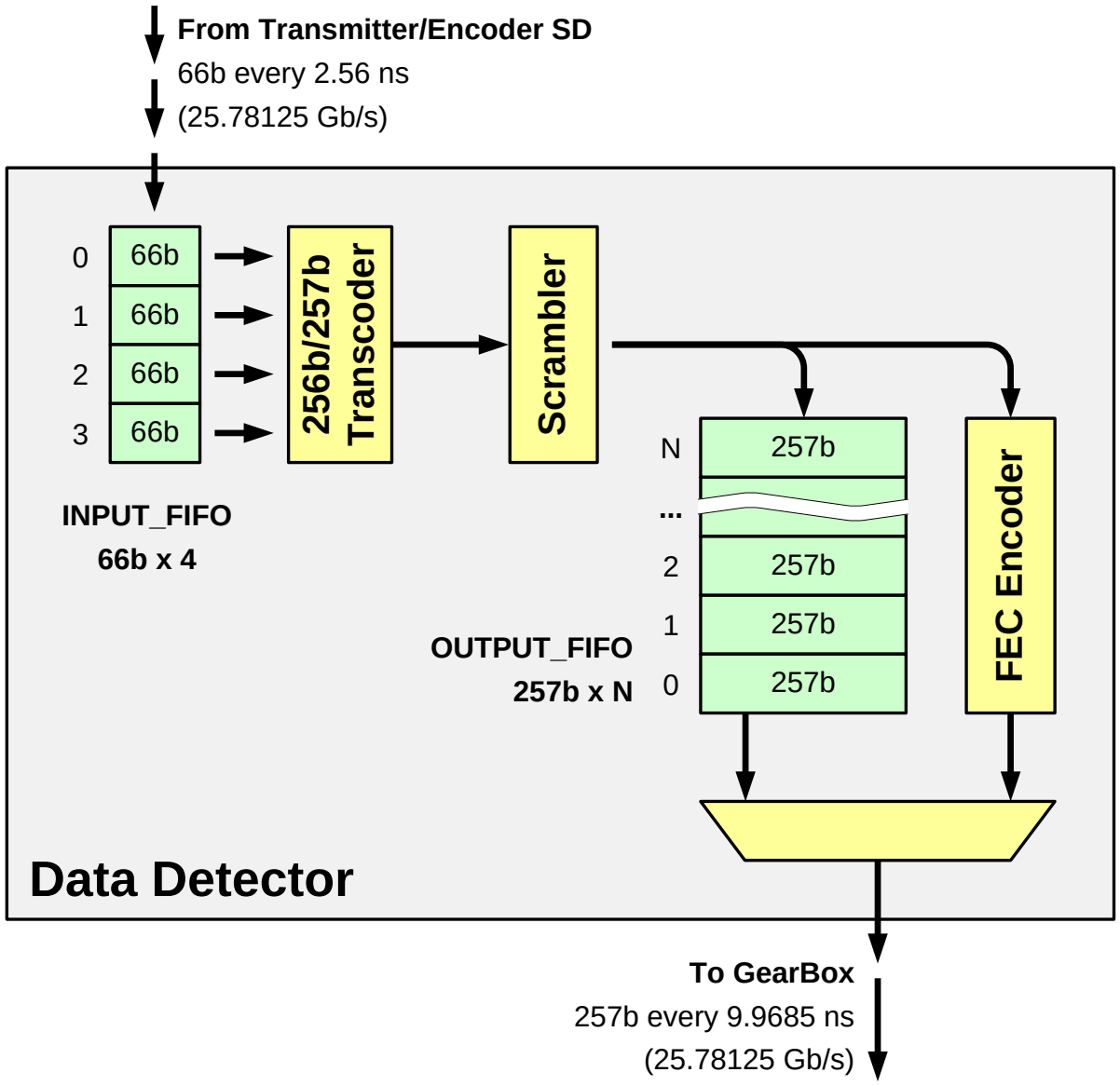


- Codeword Delimiter (CD) may be encoded as
  - A control block (block type 0x1E) and carry fixed 56-bit delimiter pattern
  - Or as an ordered set (block type 0x4B) and carry 28-bit fixed delimiter pattern plus have 24 bits reserved for physical layer signaling.

  (see backup slide #15)

Functional Diagram

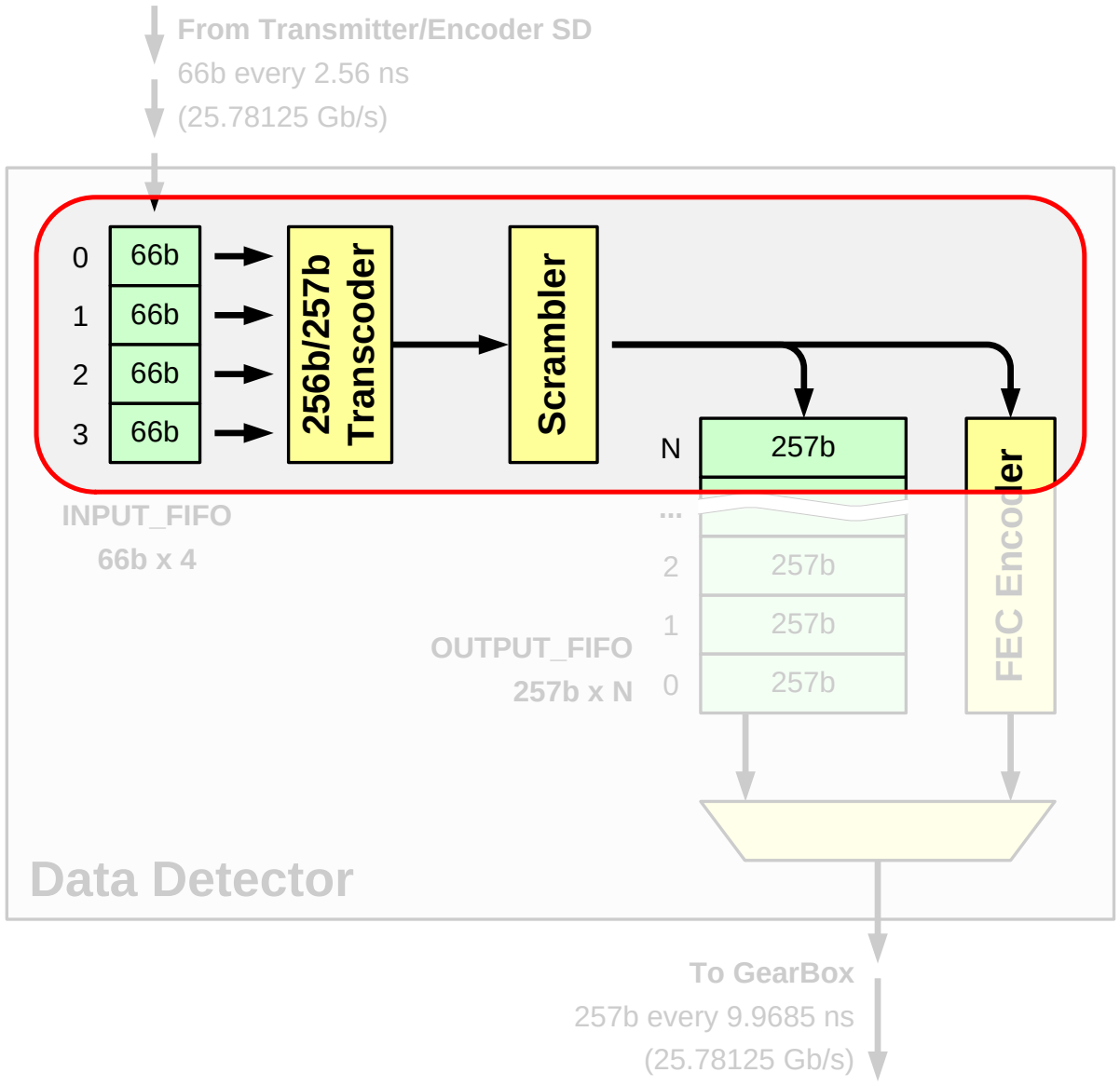
- **INPUT_FIFO** accumulates 4 66b blocks to get transcoded into one 257b block
- **OUTPUT_FIFO** holds 257b blocks long enough for
  a) the FEC Encoder to generate PARITY data
  b) In the ONU, to turn the laser on and transmit sync pattern and burst delimiter.



Data Detector – Input Process

Input Process

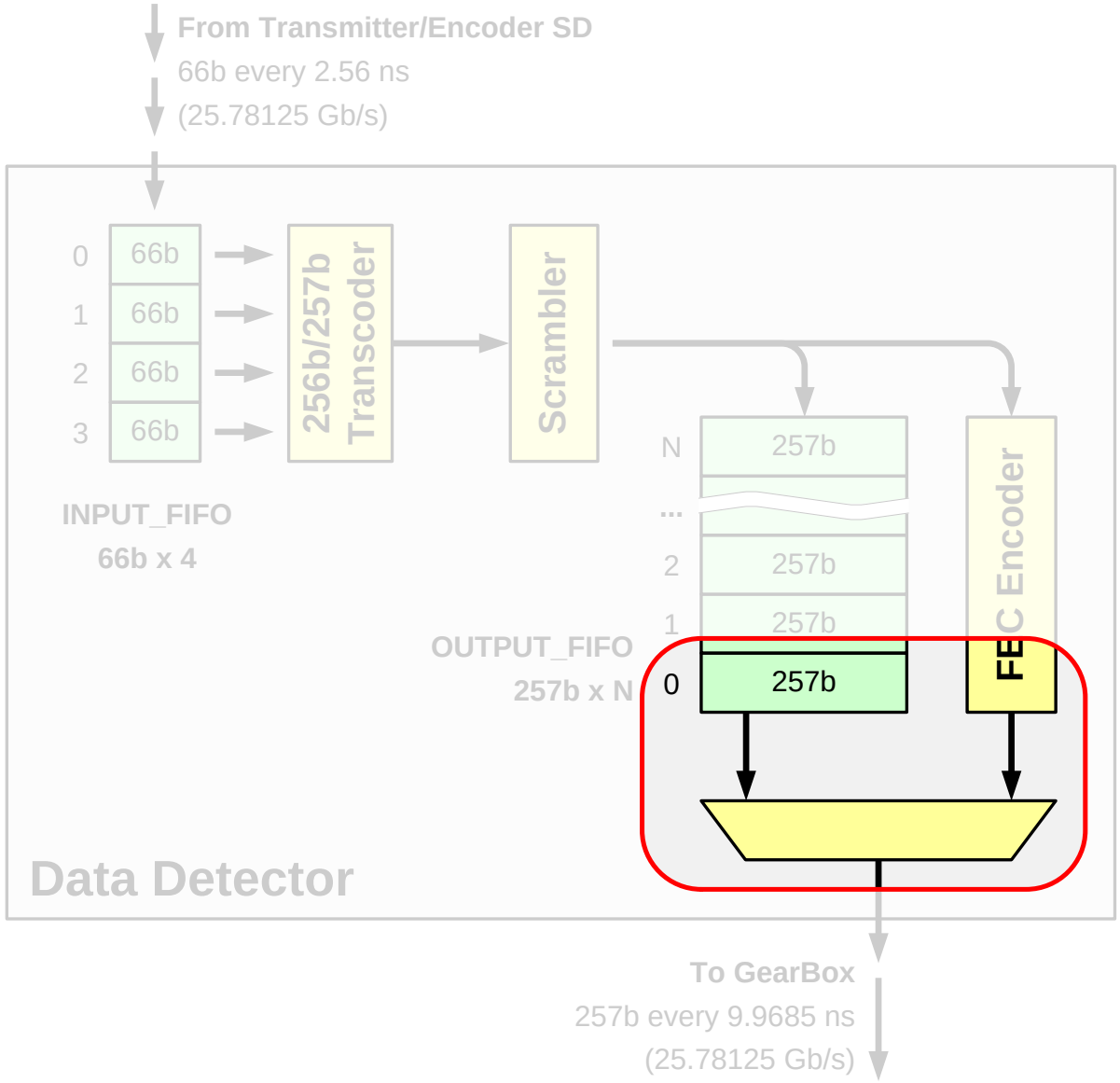
- Receives a 66-bit every 2.56 ns block from the Transmier/Encoder process
- Calls **Transcode** function on 4 66b blocks
- Calls **Scramble** function on some blocks
- Passes 257b blocks to **FEC_Encode** function
- Appends transcoded and scrambled blocks to the end of OUTPUT_FIFO



Data Detector – Output Process

Output Process

- Inserts 10 Parity blocks after every 56 payload blocks
- In the ONU,
  - Generates PMA_SIGNAL.request() to turn the laser on/off.
  - Inserts sync pattern, Burst Delimiter, and terminating sequence.
  - Handles shortened last FEC codeword.
- Passes a 257-bit block every 9.9685 ns to the GearBox
  - Output bit rate is 25.78125 Gb/s

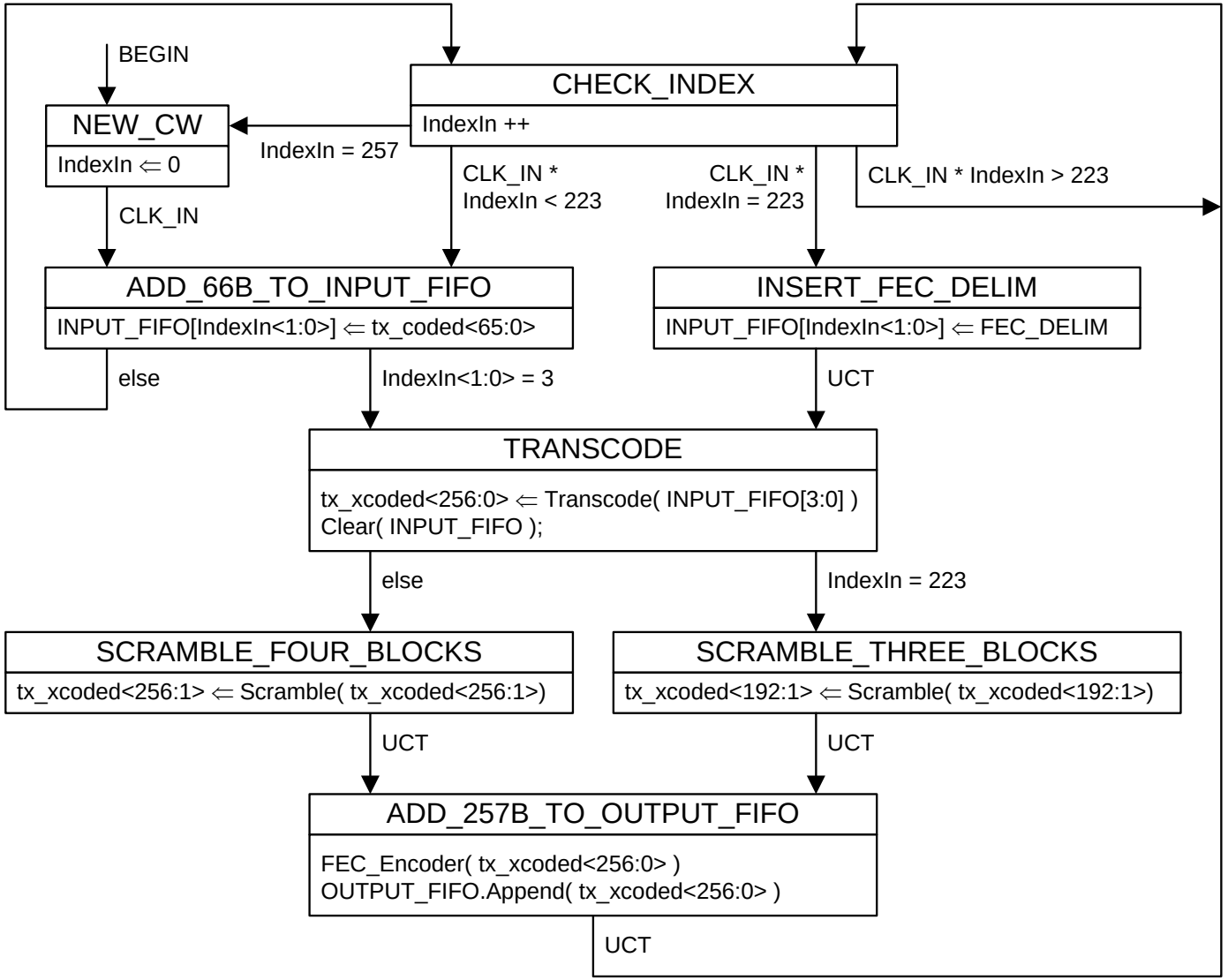


OLT PCS State Diagrams

(ONU PCS State Diagrams are coming soon)

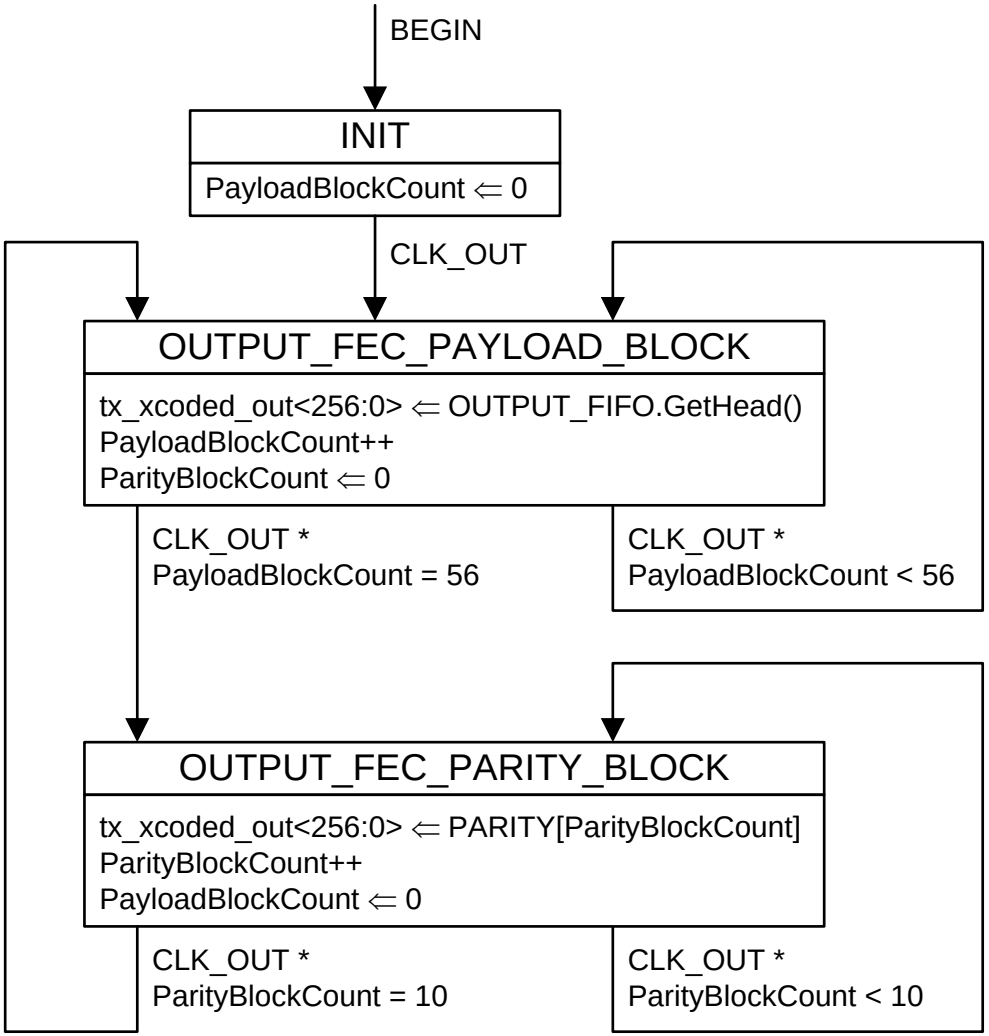
OLT Input Process State Diagram

- **CLK_IN** is the same as 25GMII clock TX_CLK25
- **IndexIn** counts the blocks received from Transmitter/ Encoder SD
  - Blocks 0-222: payload
  - Block 223: delimiter
  - Blocks 224-256: parity placeholders
- **tx_coded<65:0>** is a 66-bit vector received from Transmitter/Encoder process (see remein_3ca_1b_0118.pdf)
- **tx_xcoded<256:0>** is a transcoded 257-bit vector passed to OUTPUT_FIFO



Output Process State Diagram

- **CLK_OUT** is the rate of output of 257-bit blocks
  - One 257b sent out every 9.9685 ns
- **PayloadBlockCount** counts the payload blocks (from 0 to 56)
- **ParityBlockCount** counts the parity blocks (from 0 to 10)
- **GetHead()** returns the first element in OUTPUT_FIFO and removes it from the queue.
- **tx_xcoded_out<256:0>** is a 257-bit output vector to be passed to GearBox



Conclusion

- Proposed are simple-to-understand state diagrams for the OLT Data Detector.
  - Consists of Data Detector Input and Data Detector Output processes
  - Similar to how the DD is defined in 802.3ah and 802.3av.
- All manipulations related to burst framing and timing can be concentrated in a single state diagram (ONU Data Detector Output process).
- No need to develop any state diagrams for 256b/257b Transcoder, Scrambler, or FEC Encoder.
  - All these functions are integrated into DD Input process as function calls
  - Similar to how this is done with 64b/66b encoder in remein_3ca_1b_0118.pdf and in 802.3av.

Backup

Encoding options for FEC Delimiter

Input Data	Sync	Block Payload							
Bit Position: / Data Block Format:	0 1	2							65
$D_0\ D_1\ D_2\ D_3/D_4\ D_5\ D_6\ D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		Block Type Field							
$C_0\ C_1\ C_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0x1e	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$C_0\ C_1\ C_2\ C_3/O_4\ D_5\ D_6\ D_7$	10	0x2d	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
$C_0\ C_1\ C_2\ C_3/S_4\ D_5\ D_6\ D_7$	10	0x33	C_0	C_1	C_2	C_3		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3/S_4\ D_5\ D_6\ D_7$	10	0x66	D_1	D_2	D_3	O_0		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3/O_4\ D_5\ D_6\ D_7$	10	0x55	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
$S_0\ D_1\ D_2\ D_3/D_4\ D_5\ D_6\ D_7$	10	0x78	D_1	D_2	D_3	D_4		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3/C_4\ C_5\ C_6\ C_7$	10	0x4b	D_1	D_2	D_3	O_0	C_4	C_5	C_6	C_7
$T_0\ C_1\ C_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ T_1\ C_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ D_1\ T_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0xaa	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0\ D_1\ D_2\ T_3/C_4\ C_5\ C_6\ C_7$	10	0xb4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0\ D_1\ D_2\ D_3/T_4\ C_5\ C_6\ C_7$	10	0xcc	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0\ D_1\ D_2\ D_3/D_4\ T_5\ C_6\ C_7$	10	0xd2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0\ D_1\ D_2\ D_3/D_4\ D_5\ T_6\ C_7$	10	0xe1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0\ D_1\ D_2\ D_3/D_4\ D_5\ D_6\ T_7$	10	0xff	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

Figure 49–7—64B/66B block formats